ELONGATED BOUNDARIES DETECTOR PARAMETERS OPTIMISATION BASED ON GENERATION OF SYNTHETIC DATA FROM AERIAL IMAGERY

Ekaterina Panfilova
Evocargo LLC, Moscow, Russia
V. A. Trapeznikov Institute of
Control Sciences, RAS,
Profsoyusnaya street, 65, Moscow, 117997, Russia
E-mail: mipt.epanfilova@gmail.com

Anton Grigoryev
Evocargo LLC, Moscow, Russia
Institute for Information
Transmission Problems, RAS
Bolshoy Karetny per. 19, Moscow, 127051, Russia
E-mail: me@ansgri.com

Vladimir Burmistrov
Evocargo LLC, Moscow, Russia
E-mail: burbiksvy@gmail.com

KEYWORDS

Boundaries detector, synthetic dataset, road markings detection, Optuna toolkit

ABSTRACT

The detector of elongated boundaries in the image, such as road marking lines, rails, etc., is an important component of the visual system of a highly automated vehicle (HAV). It is used by HAV to solve self-localization problems or maintain the position inside the lane. Testing and optimisation of computer vision algorithms include preparing of the datasets that are usually labelled manually. Synthetic data of various kinds can reduce the complexity of algorithms development. In this paper we describe an approach to generation of synthetic data for testing elongated boundaries detectors that works with road images obtained from the cameras mounted on the HAV and converted to bird's eye view. This approach is based on cutting of raster aerial imagery and corresponding vector markup of target objects in the aerial imagery in various ways.

There is a class of elongated boundaries detectors, the first stage of which is the background suppression on the image and thus highlighting of the target lines. For them, we propose a method for creating a dataset consisting of images with a suppressed background, specifically, images of white elongated lines on a black background. The lines' shape will be similar to the target lines of the detector. With such a dataset you can tune the parameters, which affect stages of the algorithm, following the suppression of the background in the image.

In this paper we also consider elongated boundaries detector. Its parameters fix the geometric model of the target lines. Automatic optimisation of the quality of such a detector is possible using the Optuna toolkit, but it requires a large dataset. In this paper, we propose an approach to optimisation of the detector on a synthetic dataset. The effectiveness of this approach is confirmed by testing on real data.

INTRODUCTION

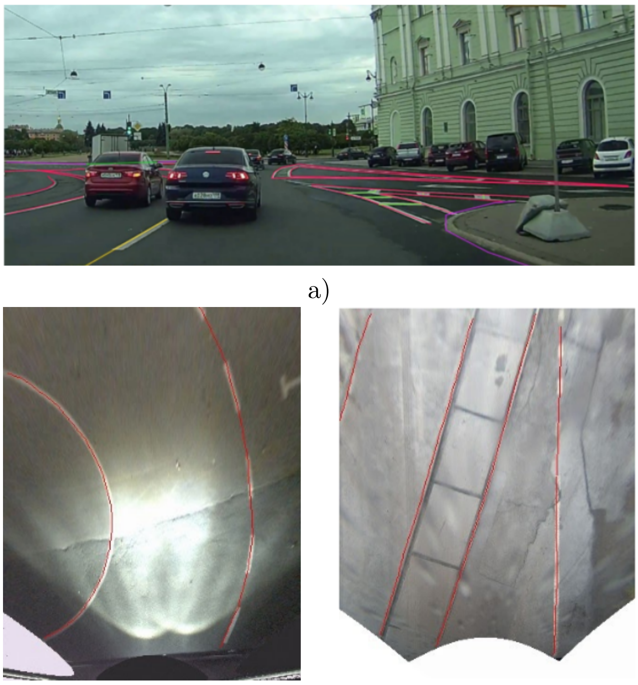
The detector of elongated boundaries in the images of the road is an important component of the visual system of a highly automated vehicle (HAV). The elongated boundaries in the images of the road can be, for example, road markings lanes, traffic-way borders, tram tracks, etc. (see fig. 1).

The information about the boundaries location in the image are necessary to solve the problems such as map-relative localization of the HAV (Ziegler et al. 2014; Shipitko and Grigoryev 2018), maintaining the position of the HAV inside the lane (Wang et al. 2005) or lane departure warnings (Jung and Kelber 2004; Hsiao et al. 2008).

There are various approaches to the detecting elongated boundaries in the world. Approaches based on machine learning methods: such as structural forest (Xiao et al. 2016), convolutional neural networks (Teplyakov et al. 2021), recurrent neural networks (Zheng et al. 2020), combinations of different types of neural networks (Dong et al. 2021; Zhang et al. 2021), neural networks which analyze not only input image but also use engineering features (Erlygin and Teplyakov 2021). Approaches based on classical image processing methods: for example, image background suppression in order to highlight target lines and then approximation of the highlighted lines by polylines using the Hough transform (Jang et al. 2014) or the window Hough transform (Panfilova and Kunina 2020). The Canny edge detector (Canny 1986) or other edge

Communications of the ECMS, Volume 36, Issue 1,
Proceedings, ©ECMS Ibrahim A. Hameed, Agus Hasan,
Saleh Abdel-Afou Alaliyat (Editors) 2022
ISBN: 978-3-937436-77-7/978-3-937436-76-0(CD) ISSN 2522-2414

detectors can also be used to hightlight target lines (Tropin et al. 2019; Mousavi et al. 2019), in combination of probabilistic Hough transform (Hou et al. 2016; Li et al. 2016) or the least squares method (Yoo et al. 2013), applied for polyline approximation.



a)

b) c)

Fig. 1: Examples of elongated boundaries in the images of the roads: a) colour highlights solid and dash-lines, traffic-way borders and safety islands, b) bird's eye view image, colour highlight solid and dash lines, c) bird's eye view image, colour highlights tram tracks and dash-lines

A dataset, usually being markuped manually, is required in order to tune the parameters and evaluate the quality of the detector, like of any other computer vision algorithm. There are datasets of roads' images obtained by a camera mounted on a vehicle: TuSimple (https://github.com/TuSimple/tusimple-benchmark), CalTech (http://www.mohamedaly.info/research/lane-detection), CuLane (Pan et al. 2018), but they are not suitable for the detectors which input data is bird's eye view image. They do not contain parameters of cameras that can be used to convert images to bird's eye view. Moreover, when HAV is launched at a certain territory, it is necessary to tune detector parameters based on data as close as possible to real one, that is, images of the area on which the HAV will move.

In this article we will present method of generating synthetic datasets consisting of bird's eye view images of the road. For the detectors, first step of which is background suppression, we propose a dataset of drawn white road marking lines on a black background. To show the practical applicability the elongated boundaries detector (Panfilova and Kunina 2020) will be optimised on the synthetic data. Than the quality of optimised parameters will be checked on a real data.

The rest of the paper is structured as follows: section II presents the algorithms of generation of synthetic dataset of two types, section III shows the practical applicability of synthetic datasets and section IV summarises the results.

SYNTHETIC DATASETS GENERATION

In this section we will present the algorithms for creating synthetic datasets of two types: aerial imagery synthetic dataset (see fig. 3) and drawn road markings synthetic dataset (see fig. 4). These datasets will be based on a raster aerial imagery (see fig. 2a) and its vector markup of target objects (see fig. 2b).

Aerial imagery synthetic dataset

This type of a dataset consists of bird's eye view images of the road with a user-specified image resolution. The line length as well as the total length of all lines in the image are limited from below. Such a dataset can be used to train and test elongated boundaries detectors that work with bird's eye view images.

Further, we will formulate the precise statement of the problem and present its solution.

Problem statement

Let we have
- Raster aerial imagery $\mathbf{I_{map}}$ size of $\mathbf{w_{map}}$ by $\mathbf{h_{map}}$
- Its markup of detector target lines (such as road markings lanes, trails, etc.) – $\mathbf{VectMarkup_{map}}$
- Image resolution – number pixels per metre is ppm_{map}

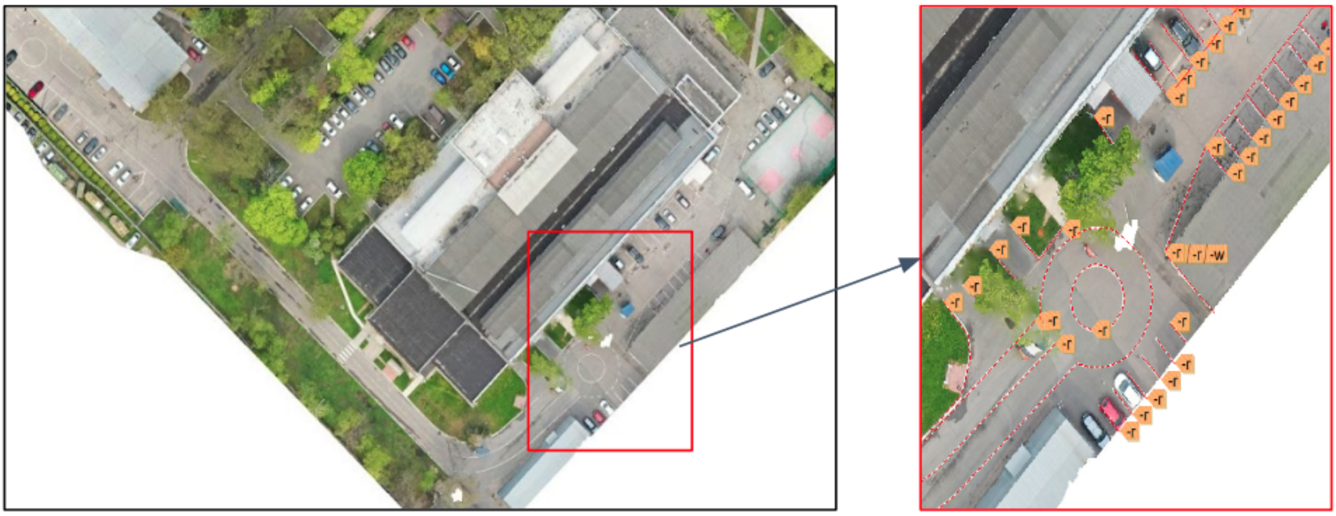
It is required to create a dataset (see fig. 3) such as:
- Bird's eye view images of the roads
- Image size is $\mathbf{w}$ by $\mathbf{h}$
- Result image resolution is $\mathbf{ppm_{im}}$
- The pixel length of each line is greater than the threshold — $\mathbf{thr_{line}}$
- The total pixel length of all lines in the image is greater than the threshold — $\mathbf{thr_{lines}}$

Algorithm of aerial imagery synthetic dataset generation

The main idea of the algorithm is to copy regions on the aerial imagery in different positions and for each position determine and save markup lines which appears in the region. The region and the obtained markup must be scaled to satisfy the requirements of result images in the datset. To obtain different positions of the region, the starting point on the aerial imagery is randomly selected, and then the region is shifted horizontally, vertically, and rotated around its centre with fixed deltas. You can see the pseudocode of this algorithm below (Algorithm 1).

The result of the algorithm 1

To obtain the aerial imagery synthetic dataset, we got an aerial imagery of a closed area "Kalibr", Moscow, Russia (a certain territory used for autonomous vehicle testing) size of $w_{map} = 6475$ by $h_{map} = 7098$



a) b)

Fig. 2: a) Fragment of aerial imagery. b) Markup of target lines in the aerial imagery fragment.

Algorithm 1 Generation of aerial imagery synthetic dataset

Input: I_{map},$VectMarkup_{map}$, ppm_{map}, w, h, ppm_{im}, sh_x, sh_y, $\Delta\alpha$, thr_{line}, thr_{lines}, $output\_path$

Output: Set of bird's eye view images size of w by h at a resolution ppm_{im} and corresponded them files of target lines markup in the $output\_path$ directory

$scale\_f \leftarrow \frac{ppm_{map}}{ppm_{im}}$
$W_w \leftarrow w * scale\_f$; $W_h \leftarrow h * scale\_f$
$x \leftarrow random(0; W_w); y \leftarrow random(0, W_y)$
ScaleParams(sh_x sh_y, thr_{line}, thr_{lines}, $scale\_f$)
while $x < w_{map} - 2 * W_w$ **do**
  while $y < h_{map} - 2 * W_h$ **do**
    $I_{wind} \leftarrow$ region on I_{map} with top left coordinate (x, y) size of $2W_w$ x $2W_h$
    $VectMarkup_{wind} \leftarrow$ objects in $VectMarkup_{map}$, which appear in I_{wind} and which length $>thr_{line}$
    for $i \leqslant \frac{360^\circ}{\Delta\alpha}$ **do**
      $\alpha = i * \Delta\alpha$
      $I_{turn} \leftarrow Turn(\alpha, x + W_w, y + W_h, I_{wind})$ - rotation around point $(x + W_w, y + W_h)$ by α
      $VectMarkup_{turn} \leftarrow Turn(\alpha, x + W_w, y + W_h, VectMarkup_{wind})$
      $I_{res} \leftarrow$ region on I_{turn} with top left coordinate $(x + \frac{W_w}{2}, y + \frac{W_h}{2})$ size of W_w x W_h
      $VectMarkup_{res} \leftarrow$ objects in $VectMarkup_{turn}$, which appear in I_{res}
      $sum\_length \leftarrow$ **SumLen**($VectMarkup_{res}$)
      if $sum\_length < thr_{lines}$ **then**
        continue
      end if
      Scale(I_{res}, $VectMarkup_{res}$, $\frac{1}{scale\_f}$)
      Save(I_{res}, $VectMarkup_{res}$, $output\_path$)
    end for
    $y = y + sh_y$
  end while
  $x = x + sh_x$
end while

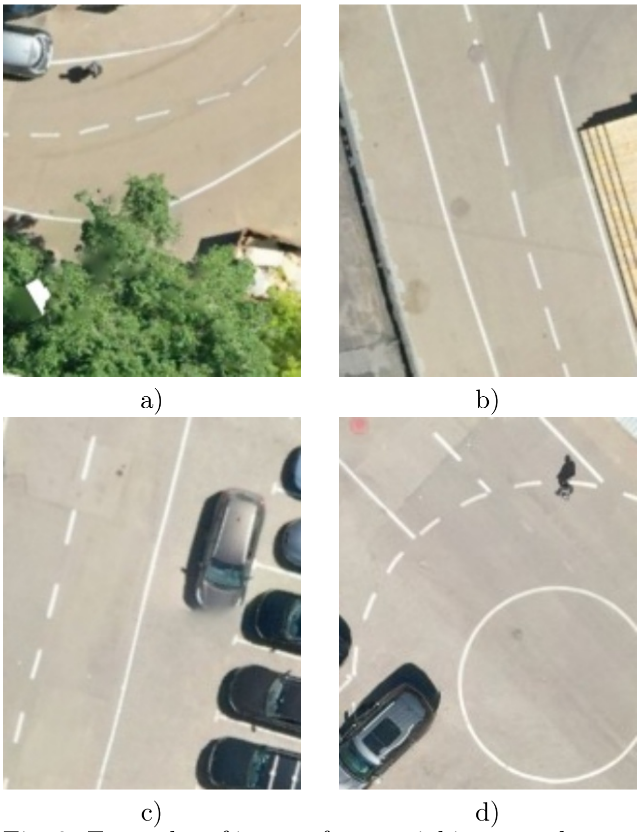


Fig. 3: Examples of images from aerial imagery dataset

pixels with image resolution $ppm_{map} = 14.5785$. Required image parameters in the dataset are $w = 320$, $h = 400$, $ppm_{im} = 60$. The given algorithm parameters are $sh_x = 160$, $sh_y = 200$, $\Delta\alpha = 60°$, $thr_{line} = 30$ pixels, $thr_{lines} = 120$ pixels. As a result we got dataset consisted of 5337 images. You can see the images examples of the resulting dataset in fig. 3, the full dataset is available at https://zenodo.org/record/6054552.

Drawn road markings synthetic dataset

This type of a dataset consists of images like white elongated lines in the road markings shape on the black background. As in previous dataset type the image resolution set by user. The line length and the total line lengths are limited from below. We can also set line thickness and the degree of image blur. Such a dateset can be used only for the road lines detectors, first step of which is background suppression. So the user can optimise the parameters that correspond the detector's steps after background suppression.

Further we will formulate the precise statement of the problem and present its solution.

Problem statement

Let us have:

• Vector marking of the detector target lines on the aerial imagery size of $\mathbf{w_{map}}$ by $\mathbf{h_{map}}$ — $\mathbf{VectMarkup_{map}}$. We will call this entity **a vector map of an aerial imagery**.

• Aerial imagery resolution - number of pixels per meter is $\mathbf{ppm_{map}}$

It is required to create a dataset such as:

• The images of white lines on a black background (see fig. 4). The lines are geometrically shaped as the target lines of the detector.

• Line thickness - $\mathbf{thk_{line}}$

• Image size is **w** by **h**

• Image resolution is $\mathbf{ppm_{im}}$

• Measure of lines blurring by Gaussian kernel size of $\mathbf{kernel_w}$ by $\mathbf{kernel_h}$ set by sigma equal to $\mathbf{sigma_x} = \mathbf{sigma_y} = \mathbf{sigma}$

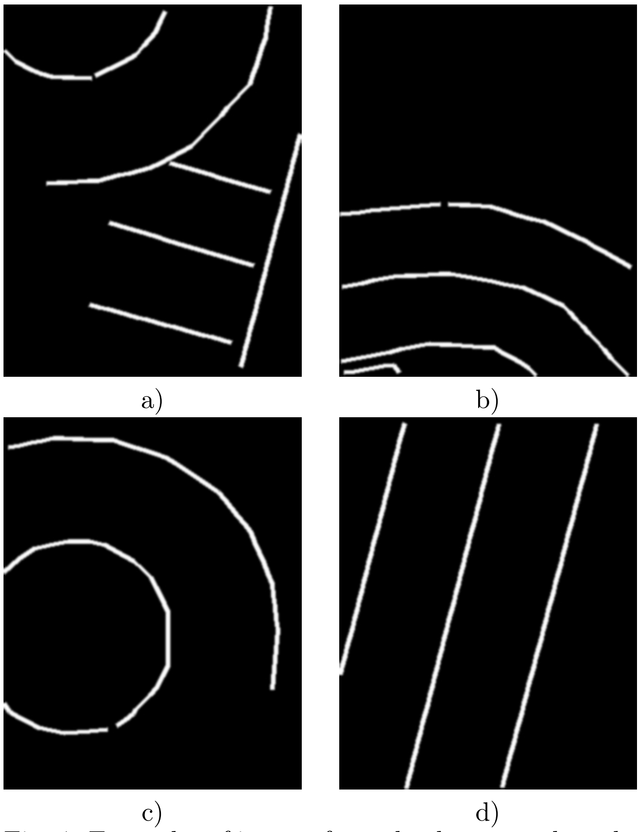


Fig. 4: Examples of images from the drawn road markings synthetic dataset

Algorithm of drawn road markings synthetic dataset generation

The main idea of the algorithm is to consider different positions of the region on the vector map of the aerial imagery. All objects of the vector map that appear into the fixed region are drawn on a black image of a given size with a certain line thickness and then blurred with a Gaussian window. To obtain different window positions, the starting point on the map is randomly selected, and then the region is shifted horizontally, vertically, and rotated around region centre by fixed deltas. You can see the pseudocode of this algorithm below (Algorithm 2).

The result of the algorithm 2

To obtain synthetic dataset we took the same as for the first algorithm aerial imagery of closed territory "Kalibr". So parameters of the aerial imagery are

Algorithm 2 Generation of drawn road markings synthetic dataset

Input: $VectMarkup_{map}$, ppm_{map}, w, h, ppm_{im}, sh_x, sh_y, $\Delta\alpha$, thr_{line}, thr_{lines}, thk_{line}, $kernel_w$, $kernel_h$, $sigma$, $output\_path$

Output: Set of images of white lines on a black background and corresponded them files of target lines markup in the $output\_path$ directory

$scale\_f = \frac{ppm_{map}}{ppm_{im}}$
$W_w \leftarrow w * scale\_f$; $W_h \leftarrow h * scale\_f$
$x \leftarrow random(0; W_w); y \leftarrow random(0, W_y)$
ScaleParams(sh_x sh_y, thr_{line}, thr_{lines}, $scale\_f$)
while $x < w_{map} - 2 * W_w$ **do**
  while $y < h_{map} - 2 * W_h$ **do**
    $I_{wind} \leftarrow$ region with top left coordinate (x, y) size of $2W_w$ x $2W_h$
    $VectMarkup_{wind} \leftarrow$ objects in $VectMarkup_{map}$, which appear in I_{wind} and which length $> thr_{line}$
    for $i \leqslant \frac{360^\circ}{\Delta\alpha}$ **do**
      $\alpha = i * \Delta\alpha$
      $VectMarkup_{turn} \leftarrow Turn(\alpha, x + W_w, y + W_h, VectMarkup_{wind})$ - rotation around $(x + W_w, y + W_h)$
      $I_{turn} \leftarrow$ region with top left coordinate $(x + \frac{W_w}{2}, y + \frac{W_h}{2})$ size of W_w x W_h
      $VectMarkup_{res} \leftarrow$ objects in $VectMarkup_{turn}$, which appear in I_{turn}
      $sum\_length \leftarrow$ **SumLen**($VectMarkup_{res}$)
      if $sum\_length < thr_{lines}$ **then**
        continue
      end if
      Scale($VectMarkup_{res}$, $\frac{1}{scale\_f}$)
      $I_{res} \leftarrow$ 3-channel RGB image size of w by h filled with $(0, 0, 0)$ pixels – black colour
      $clr \leftarrow (255, 255, 255)$ – set white colour
      Draw(I_{res}, $VectMarkup_{res}$, thk_{line}, clr)
      GaussianBlur(I_{res}, $kernel_w$, $kernel_h$, $sigma$)
      Save(I_{res}, $VectMarkup_{res}$, $output\_path$)
    end for
    $y = y + sh_y$
  end while
  $x = x + sh_x$
end while

the same. The required properties of the image are $w = 320$, $h = 400$, $ppm_{im} = 60$. The given algorithm parameters are $sh_x = 320$, $sh_y = 400$, $\Delta\alpha = 120$, $thk_{line} = 5$, $kernel_x = kernel_y = 7$, $sigma = 1$, $thr_{line} = 30$ pixels, $thr_{lines} = 120$ pixels. As a result we got a dataset consisted of 661 images. The images example you can find in the figure 4. The full dataset is avalaible at https://zenodo.org/record/6054552.

EXPERIMENTAL RESULTS

Elongated boundaries detector

In order to test the practical applicability of the synthetic dataset, we considered the elongated boundaries detector from the article (Panfilova and Kunina 2020). This detector accepts as input a bird's eye view image. Its first stage is the suppression of the background of the image and thus the highlighting of the target lines. The second stage is the approximation of the highlighted lines by polylines and filtration them by length. The quality of this algorithm was tested on the open dataset (see Fig. 3 (a)) presented in the proceedings (Panfilova et al. 2021). The dataset consists of road images from the the closed territory "Kalibr", Moscow, Russia, files with target lines markup and the parameters of the camera on which the dataset was collected. Knowing the camera's parameters we can convert images to bird's eye view (see fig. 5). The algorithm parameters were tuned by an expert. The found line in the image was considered TP if its segments were close enough to the target line and coincided with the target line direction (see proceedings (Panfilova et al. 2021) for a more detailed description of the quality metrics). Precision of the algorithm is 0.43, recall - 0.73, F-score - 0.54.

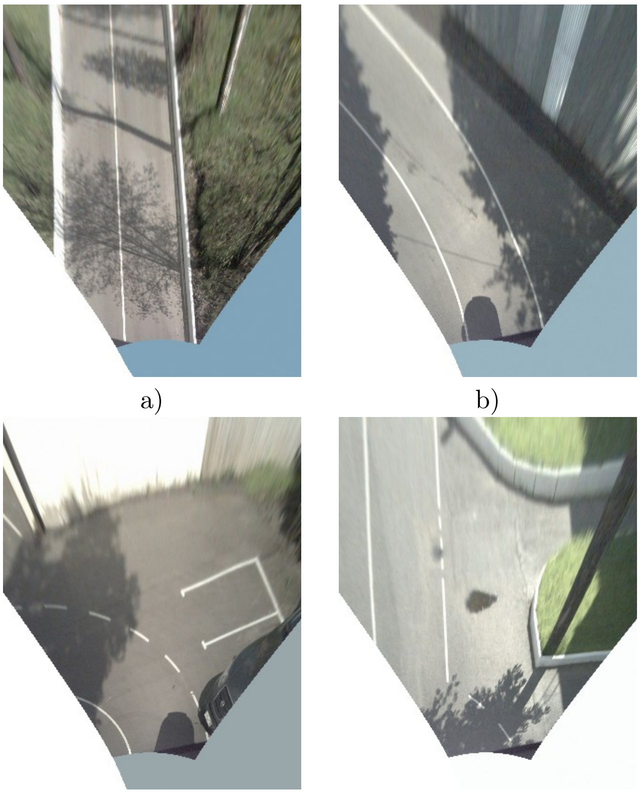


Fig. 5: Examples of images from the open dataset (Panfilova et al. 2021) converted to bird's eye view

Optuna optimisation using synthetic dataset.

In the second section we presented two synthetic datasets: aerial imagery (see fig. 3) and drawn road markings (see fig. 4). In order to optimise the detector parameters and evaluate the detector we merged these datasets together and randomly split merged data into train and evaluation sub-samplings. Train sub-sampling consists of 1300 images: 1000 images of aerial imagery dataset and 300 images of drawn markings line dataset. Test sub-sampling consists of 4998 images: 4337 images of aerial imagery dataset and 661 images of drawn road markings dataset.

To optimise the detector parameters we used Optuna toolkit (Akiba et al. 2019). It allows to optimise model based algorithms like the detector (Panfilova and Kunina 2020). The target value was F-score. We ran the optimisation of the detector on the train subsampling of the synthetic dataset 6 times to get the F-score distribution with randomly chosen starting parameters and with best parametrs after optimisation. The distributions were calculated for the train and test sub-samplings of synthetic dataset and for the open dataset of real data obtained from the same territory as aerial imagery for the synthetic dataset (Panfilova et al. 2021). As a result we got the quality of the detector

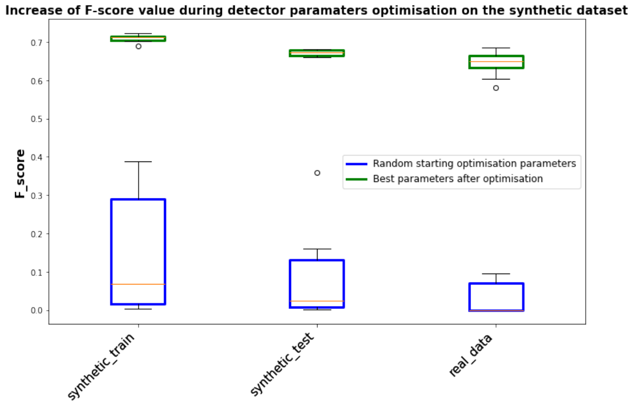


Fig. 6: Boxplot of F-score distribution

on the real data with optimised parameters on average 0.54 higher than the quality of one with randomly chosen starting parameters for optimisation process. You can see the visualisation of the obtained results in the fig. 6.

CONCLUSIONS

In this paper we proposed the approach to generation of synthetic datasets of two types: aerial imagery dataset (see fig. 3) and drawn road markings on a black background dataset (see fig. 4). The main advantage of this method is that you only need aerial imagery and you have to markup the target lines on it just once. The aerial imagery dataset intended to be used for train and test elongated boundaries detectors which input image is a bird's eye view image of the road. The drawn markings dataset was proposed for training the detectors first stage of which is background suppression and whose input image is also a bird's eye view image of the road. In this case you will be able to optimise not all detector's parameters but those that correspond the stages followed by the background suppression. The generated sythetic datasets from the aerial imagery of the "Kalibr", Moscow, Russia are available at https://zenodo.org/record/6054552.

Moreover, we showed the practical applicability of the synthetic datasets. Considering the elongated boundaries detector from the proceedings (Panfilova and Kunina 2020), we optimised it on the train part of synthetic dataset and show the quality gain on a real data. Thus, using synthetic data can reduce time of data collecting and it markup and gives acceptable detector quality on a real data.

Further improvement of the synthetic data generation may be focused on simulating different weather conditions, like puddles, shadows, wet paved road and etc.

REFERENCES

Akiba, Takuya; Shotaro Sano; Toshihiko Yanase; Takeru Ohta; and Masanori Koyama. 2019. "Optuna: A next-generation hyperparameter optimization framework." In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Canny, John. 1986. "A computational approach to edge detection." *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Dong, Yongqi; Sandeep Patil; Bart van Arem; and Haneen Farah. 2021. "A hybrid spatial-temporal deep learning architecture for lane detection." *arXiv preprint arXiv:2110.04079*.

Erlygin, L. A. and L. M. Teplyakov. 2021. "Improvement of a line segment detector based on a neural network by adding engineering features." *Sensory systems*, 35(1):50–54. DOI: 10.31857/S0235009221010042.

Hou, Changzheng; Jin Hou; and Chaochao Yu. 2016. "An efficient lane markings detection and tracking method based on vanishing point constraints." In *2016 35th Chinese Control Conference (CCC)*, 6999–7004.

Hsiao, Pei-Yung; Chun-Wei Yeh; Shih-Shinh Huang; and Li-Chen Fu. 2008. "A portable vision-based real-time lane departure warning system: day and night." *IEEE Transactions on Vehicular Technology*, 58(4):2089–2094.

Jang, Ho-Jin; Seung-Hae Baek; and Soon-Yong Park. 2014. "Lane marking detection in various lighting conditions using robust feature extraction."

Jung, Claudio Rosito and Christian Roberto Kelber. 2004. "A lane departure warning system based on a linear-parabolic lane model." In *IEEE Intelligent Vehicles Symposium, 2004*, 891–895.

Li, Yadi; Liguo Chen; Haibo Huang; Xiangpeng Li; Wenkui Xu; Liang Zheng; and Jiaqi Huang. 2016. "Nighttime lane markings recognition based on canny detection and hough transform." In *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 411–415.

Mousavi, Seyed Muhammad Hossein; Vyacheslav Lyashenko; and Surya Prasath. 2019. "Analysis of a robust edge detection system in different color spaces using color and depth images." *Компьютерная оптика*, 43(4):632–646.

Pan, Xingang; Jianping Shi; Ping Luo; Xiaogang Wang; and Xiaoou Tang. 2018. "Spatial as deep: Spatial cnn for traffic scene understanding." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Panfilova, E. I. and I. A. Kunina. 2020. "Using window hough transform for detecting elongated boundaries in an image." *Sensory systems*, 34(4):340–353. DOI: 10.31857/S0235009220030075.

Panfilova, Ekaterina; Oleg S Shipitko; and Irina Kunina. 2021. "Fast hough transform-based road markings detection for autonomous vehicle." In *Thirteenth International Conference on Machine Vision*, volume 11605, page 116052B.

Shipitko, Oleg and Anton S Grigoryev. 2018. "Ground vehicle localization with particle filter based on simulated road marking image." In *ECMS*, 341–347.

Teplyakov, Lev; Kirill Kaymakov; Evgeny Shvets; and Dmitry Nikolaev. 2021. "Line detection via a lightweight cnn with a hough layer." In *Thirteenth International Conference on Machine Vision*, volume 11605, page 116051B.

Tropin, D. V.; Y. A. Shemyakina; I. A. Konovalenko; and I. A. Faradjev. 2019. "Localization of planar objects on the images with complex structure of projective distortion." *Informatsionnye protsessy*, 19(2):208–229.

Wang, Jin; Stefan Schroedl; Klaus Mezger; Roland Ortloff; Armin Joos; and Thomas Passegger. 2005. "Lane keeping based on location technology." *IEEE Transactions on Intelligent Transportation Systems*, 6(3):351–356.

Xiao, Liang; Chuanxiang Li; Dawei Zhao; Tongtong Chen; and Bin Dai. 2016. "Road marking detection based on structured learning." In *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, 2047–2051.

Yoo, Hunjae; Ukil Yang; and Kwanghoon Sohn. 2013. "Gradient-enhancing conversion for illumination-robust lane detection." *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1083–1094.

Zhang, Ce; Yu Han; Dan Wang; Wei Qiao; and Yier Lin. 2021. "A network that balances accuracy and efficiency for lane detection." *Mobile Information Systems*, 2021.

Zheng, Tu; Hao Fang; Yi Zhang; Wenjian Tang; Zheng Yang; Haifeng Liu; and Deng Cai. 2020. "Resa: Recurrent feature-shift aggregator for lane detection." *arXiv preprint arXiv:2008.13719*, 5(7).

Ziegler, Julius; Henning Lategahn; Markus Schreiber; Christoph G Keller; Carsten Knöppel; Jochen Hipp; Martin Haueis; and Christoph Stiller. 2014. "Video based localization for bertha." In *2014 IEEE intelligent vehicles symposium proceedings*, 1231–1238.

AUTHOR BIOGRAPHIES

EKATERINA PANFILOVA was born in Moscow, Russia. She studied applied physics and mathematics, and obtained her Master degree in 2019 in Moscow Institute of Physics and Technology. Currently she is a Phd. student in V. A. Trapeznikov Institute of Control Science, RAS. Since 2018, she has been working as a junior researcher at the Vision Systems Lab of the Institute for Information Transmission Problems and since 2021 - at Evocargo LLC as a software engineer. Her research activities focus on the areas of computer vision and image processing.
Her email address is mipt.epanfilova@gmail.com.

VLADIMIR BURMISTROV was born in Kiev, Ukraine. Currently he is pursuing a bachelor's degree in computer science and computer technology at the Higher School of Economics. Since 2020, he has been working in Evocargo LLC as a software engineer. His research interests are focused on robotics and data pipelines architecture.
His e-mail address is burbiksvy@gmail.com.

ANTON GRIGORYEV was born in Petropavlovsk-Kamchatskiy, Russia. Having graduated from Moscow Institute of Physics and Technology, he has been developing industrial computer vision systems with the Vision Systems Lab at the Institute for Information Transmission Problems since 2010. Currently he is also working at Evocargo LLC as a leading software engineer. His research interests are image processing and enhancement methods, autonomous robotics and software architecture.
His e-mail address is me@ansgri.com.